

Integration document



Summary

EVOLUTIVE TABLE.....	ERRO! INDICADOR NÃO DEFINIDO.
1. GLOSSARY	4
2. INTRODUCTION	5
3. KNOWING THE ENVIRONMENT CORREIOS API.....	ERRO! INDICADOR NÃO DEFINIDO.
4. REQUISITS TO USE CORREIOS API.....	ERRO! INDICADOR NÃO DEFINIDO.
5. ACCESS TO THE APIS.	7
A. MAKING THE FIRST TESTS <u>IN HOMOLOGATION</u>	9
6. API CHARACTERISTICS.....	13
7. CORREIOS' APIS.....	14
8. PREPARING THE ENVIRONMENT ON POSTMAN	15
B. CREATING A COLLECTION	16
C. CREATING THE REQUISITION	ERRO! INDICADOR NÃO DEFINIDO.
D. KNOWING THE SCREEN.....	ERRO! INDICADOR NÃO DEFINIDO.
E. MAKING THE FIRST REQUEST THROUGH POSTMAN	19
9. API CEP (ZIP CODE API)	21
10. API AGÊNCIA (AGENCY/POST OFFICE API)	23
11. API PREÇO (PRICE API) AND API PRAZO (DELIVERY TIME API).....	26
12. API RASTRO (TRACKING API)	34
13. FINAL CONSIDERATIONS.....	ERRO! INDICADOR NÃO DEFINIDO.

Evolutionary table

Date	Improvement	Version
03/18/2022	Creation	1.0

1. Glossary

In order to facilitate understanding, we have listed below a small glossary with the main terms related to pre-postage.

Term	Description
Authentication	Process of verifying a user's digital identity, which means, to ensure that the user is really who he says he is.
Authorization	Process of verifying which are the given privileges to use an application.
Postage card	Card that identifies a user as a Correios Contract customer.
Service Code	Code of the services contracted by the customer along with Correios. The contracted service codes are included in the commercial contract that the customer has signed with Correios.
Commercial Contract	Document that the customer signs with the Correios for the provision of services of parcels, messages, direct marketing and others.
Meu Correios (My Correios)	Authentication and authorization mechanism for accessing the services that Correios makes available through the internet.
Component's Password	Meu Correios' user-defined password that allows a user to use the components provided by Correios.
Token – Api Manager	Token generator system for API access. Generates an encoded string in order to ensure the security of whoever is using the API and to prevent fraud.

2. Introduction

This material was developed to contribute to the planning and integration with Correios in any platform desired by the customer.

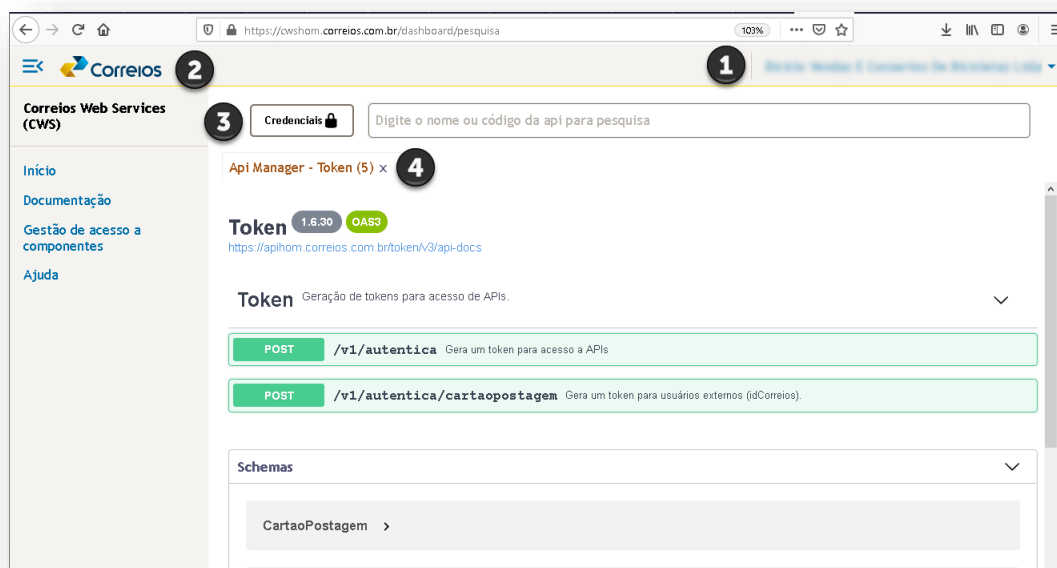
For those who want to integrate with Correios, we present the [Correios API](#), that shows which components are released for use by the customer. The components covered in this material are APIs, developed in REST and using the HTTP protocol. Which means, a base url will be provided and the HTTP verbs will indicate which action is being requested by the client.

Finally, the access to the [Correios API](#) is one of the benefits of Correios' commercial policy.

3. Knowing the Correios API environment

The access to APIs must be performed through the Correios API catalog. In the catalog the customer will be able to test and follow the changes when they happen. The catalog is available in the Homologation and Production environment:

- Homologation: <https://cwshom.correios.com.br/>
- Production: <https://cws.correios.com.br/>



Knowing the Correios Web Services' screen:

1. Client's name, option's list to access Meu Correios, exit environment and other configurations;
2. Hamburger button, opens the lateral menu and allows you to access the documentation and component's access management;
3. Search bar and credentials generator button, the search bar shows a component's list that you are allowed to access. The credentials button allows you to generate the token to do the request's attempt through the site;
4. Component tab, the content of each bar shows the API name, version and a link to the api-docs. Just below the title, or the group, shows the verb of the request. Clicking on the verb "POST" opens a place where tests can be carried out by the browser using the "Try-it out" functionality.

4. Requisites to use Correios API

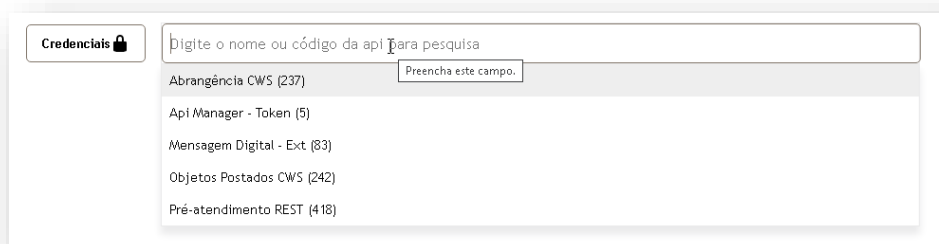
- Must have a valid registration in the [Meu Correios](#);
- To access the API, you use a password defined for the component, to create the password you must access the [Correios API](#);
- Must have an active contract for the provision of parcel or message services with the Post Office;
- Companies that provide IT solutions in the logistics segment. (How to identify and provide access?)

5. Access to the APIs.

The Correios' APIs come with access rules as follows:

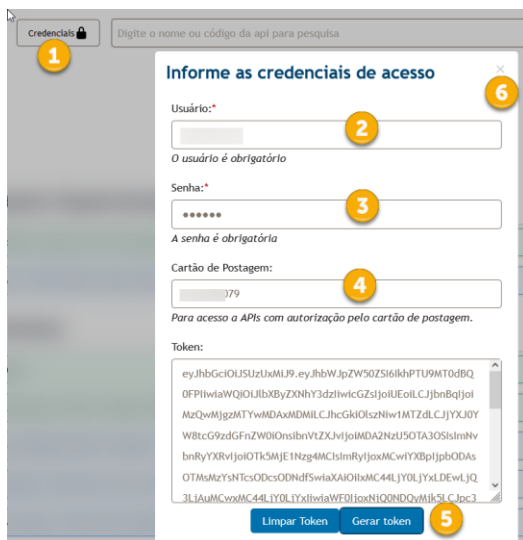
- Public API: Where everybody sees the API;
- Restrict API: The access is made from the service registration in the postage card on.

Each contract has its particularity to better serve each client. In order to take a look at the released APIs on your profile, simply click on the search bar and see what is available.



As the same as the CEP API, it is not visible in the list because it is a restricted API, which means, it requires a registration of the service in the post card.

In order to make the restricted APIs visible, you must include the data as below:



The screenshot shows a web form titled 'Informe as credenciais de acesso'. It contains the following fields and elements:

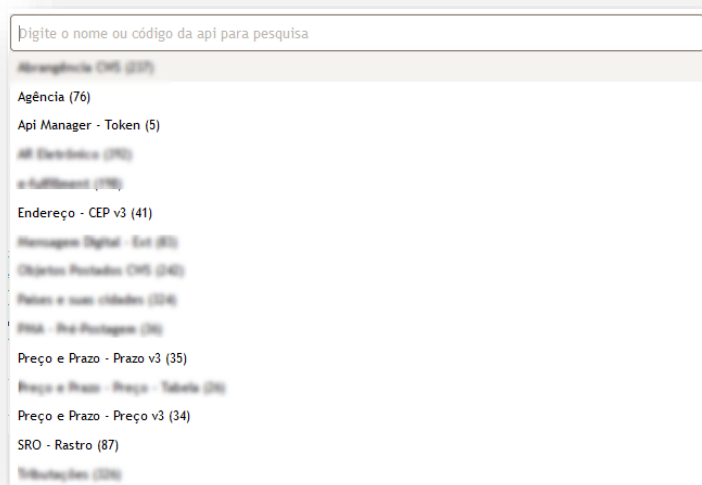
- 1**: Tab labeled 'Credenciais'.
- 2**: 'Usuário:' input field.
- 3**: 'Senha:' input field with masked characters.
- 4**: 'Cartão de Postagem:' input field with the value '779'.
- 5**: 'Gerar token' button.
- 6**: Close button (X) in the top right corner.

Below the 'Cartão de Postagem' field, there is a 'Token:' section displaying a long alphanumeric string.

1. **User credentials,**
2. **Password,**
3. **Postage card,**
4. **Generate Token and**
5. **Close.**

See that now the following APIs appears:

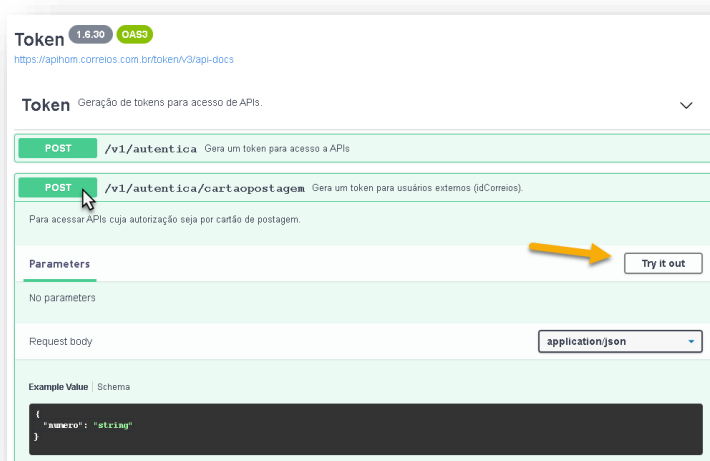
- Agência (Agency);
- Endereço – CEP v3 (Address – CEP v3);
- Prazo (Deadline);
- Preço (Price);
- Rastro (Track).



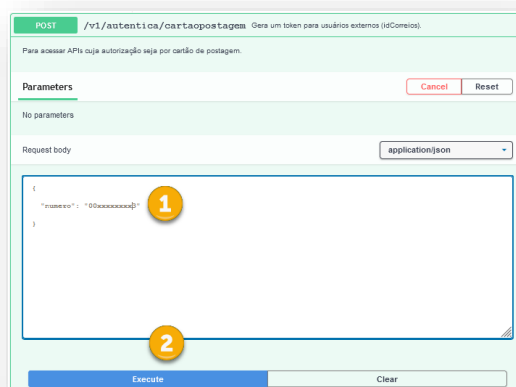
A. Making the first test in homologation

For the first test, it will be with the generation of the Token, an API without restriction and necessary in the generation of an encoded string with the authentication and authorization data to inform the APIs of the Correios.

When click on “POST”, it will appear the sub-item *Parameters* and to enable the environment, just click on “*Try it out*” button.



After clicking at the button, the screen will allow you to input the postage card (1) and click at “execute”.



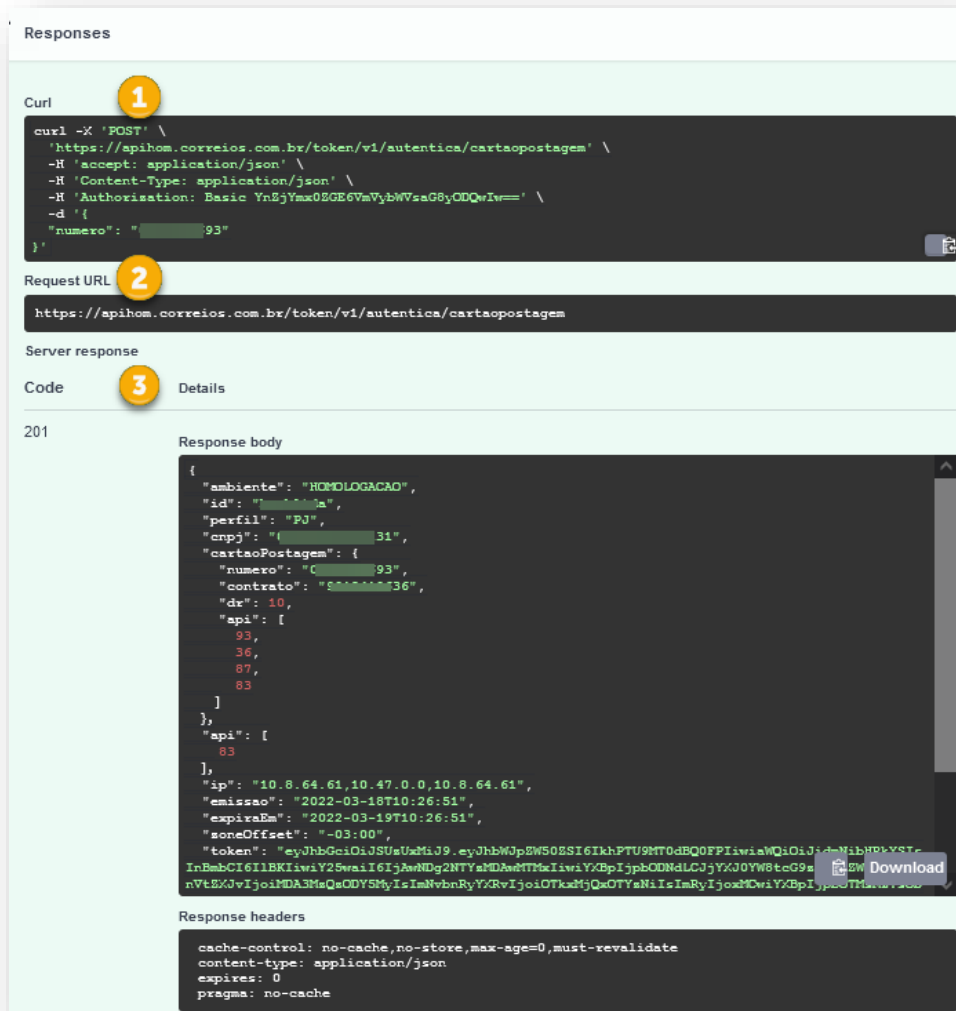
After execute, there will be a server answers, just roll the page down, until the sub-item *Responses*.

On item 1: Curl used commands or commands script, among it is the requisition verb,

at this case, “POST”;

On item 2: Requisition URL;

On item 3: Code 201 – Created, where the return Body contains the return in JSON.



On the token's requisition body we will have: environment, id, profile, contract's data, token's expiration date and the token.

Be mindful to the token's expiration date.

See the return of request example:

Data related to the postage card	{ "ambiente": "HOMOLOGACAO",
---	---------------------------------

	<pre> "id": "xxxxxxxxxx", "perfil": "PJ", "cnpj": "00000000000101", "cartaoPostagem": { "numero": "00xxxxxxx7", "contrato": "99xxxxxxx", "dr": 10, "api": [93, 36, 83] }, "api": [93, 36, 54, 87], </pre>
Token's expiration date ("expiraEm")	<pre> "emissao": "2022-03-18T10:26:51", "expiraEm": "2022-03-19T10:26:51", </pre>
Token	eyJhbGciOiJSUzUxMiJ9.....

Exploring the bottom of the Token tab page, we will find the Schemas Group, where you can view the properties.



6. API Characteristics

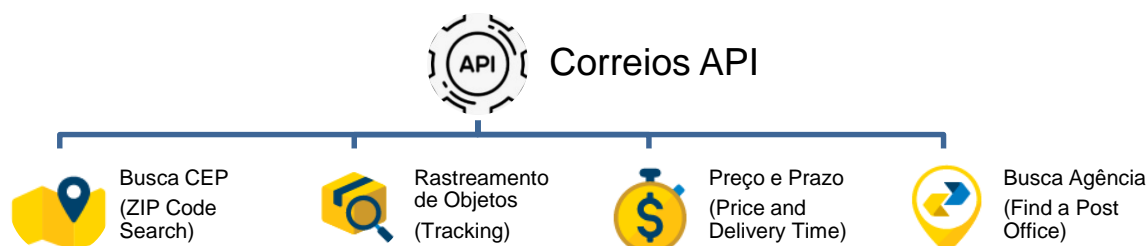
The Correios APIs was developed with REST technology, besides of being a market standard, it allows you to choose or adopt which programming language you can use, such as: ASP, .Net, Java, PHP, Ruby, Python, among others.

Among other characteristics, the features that stand up the most at API are:

- **Absence of proprietary applications:** There is no need to install a specific environment to use the API.
- **Simplicity:** The protocol used is purely the HTTPS.
- **Credentials:** The treatment of client's credentials transit in a secure environment.

7. Correios APIs

The Correios API is a micro-services-oriented architecture allowing to develop systems that are more flexible, scalable and with simpler maintenance.



As the volume of requests may vary from 1 or hundreds of requests, to obtain a quick response from the Correios server we are adopting two forms of request: the synchronous form and the asynchronous form.

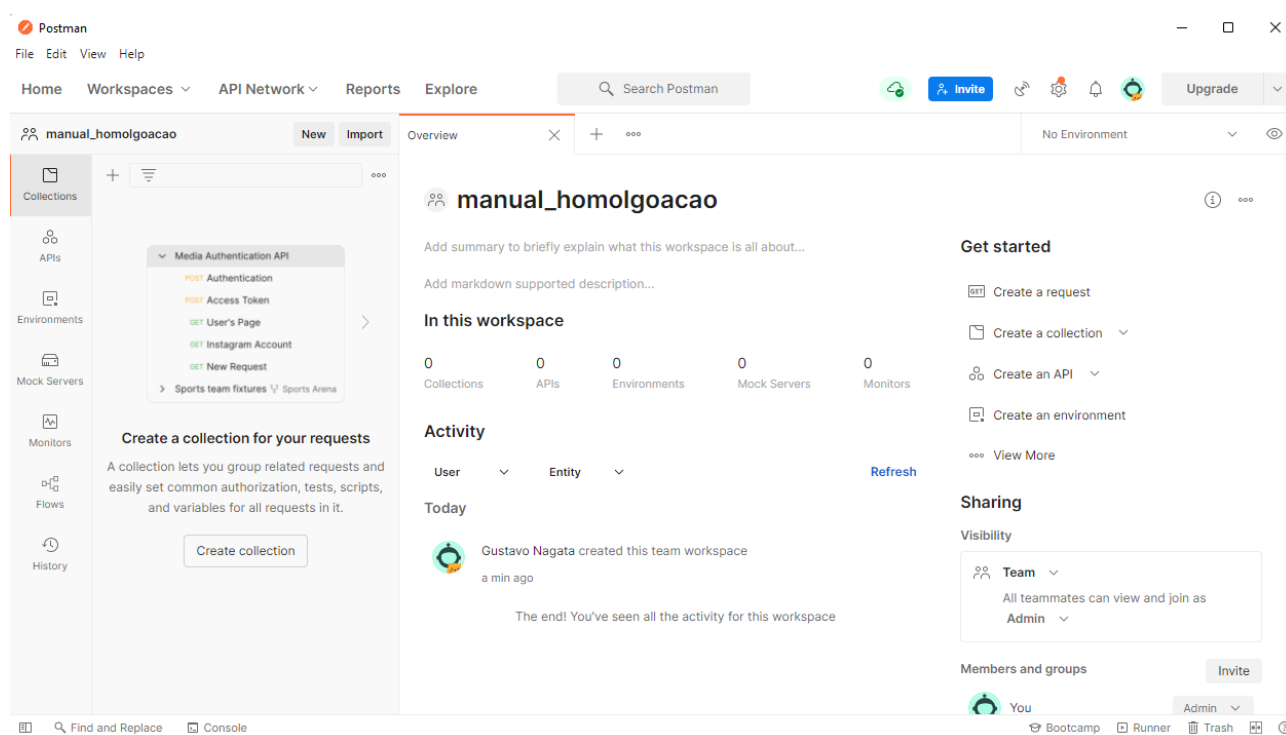
The difference between synchronous and asynchronous:

- A synchronous request must wait for the completion of processing and can only make a new request after the request is answered, this action may give the feeling of freezing the request.
- In an asynchronous request, the client sends a request and receives data that allows querying the processing status. This allows to perform new asynchronous requests, without the need to wait for the other requests to finish processing. At the end, you will be able to request the collection of the data that has been processed.

8. Preparing the environment on Postman

To start the tests, without a tool in which you can save what was developed, it can generate rework. For this purpose, in this manual, we will use the Postman tool that allows us to carry out the tests and, most importantly, save the work.

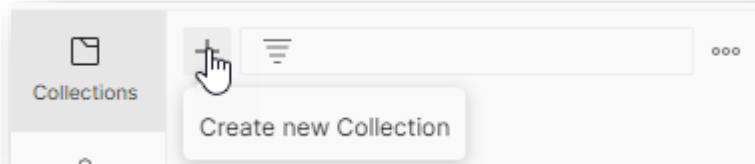
In addition to [downloading](#) the tool, you can use an extension through the Google Chrome browser, adding the Postman extension. There are other tools, such as [Insomnia](#). You can use the tool of your choice.



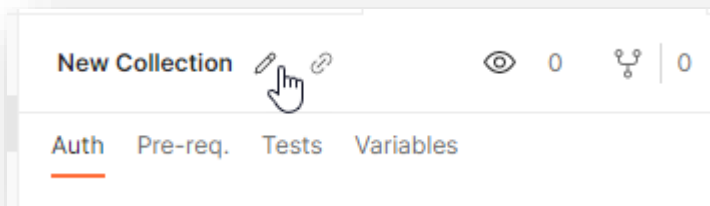
We clarify that the adoption of Postman is for educational purposes only, considering that it facilitates the understanding of the steps to access the APIs provided by Correios.

B. Creating a collection

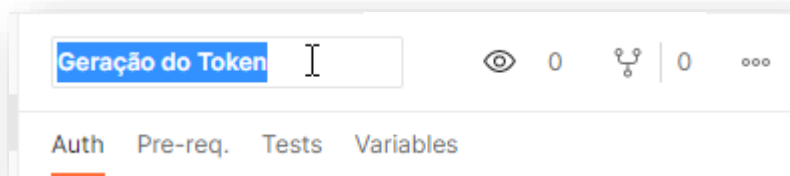
The collection allows you to separate the API that will be used throughout the manual, which can be used to separate the process. Such as: Token Generation, Pre-Post Generation, Label Generation, etc. Click on the “+” and a collection will appear that says "New Collection".



Click on “New Collection” and, at the right side, click on the pencil icon, at the right side of the name.

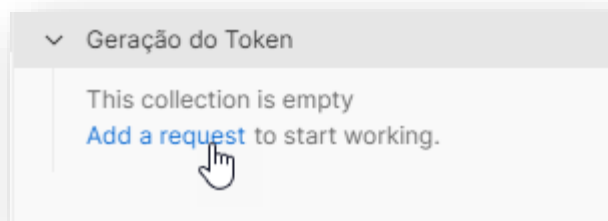


Write the collection's name, for instance: “Geração do Token” (Token Generation).

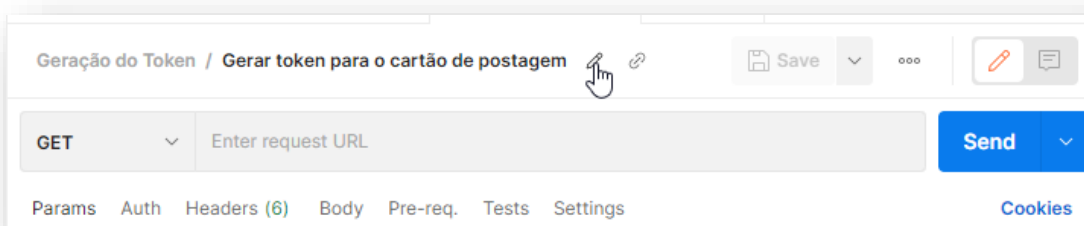


C. Creating the request

To create a request, after creating a collection, click on “Add a request”.



At the right side, a tab will open, where you could rename it to “Gerar token para o cartão de postagem” (Generate token for the postage card)



Generate token for the postage card

D. Knowing the screen

Item 1: Navigation tabs;

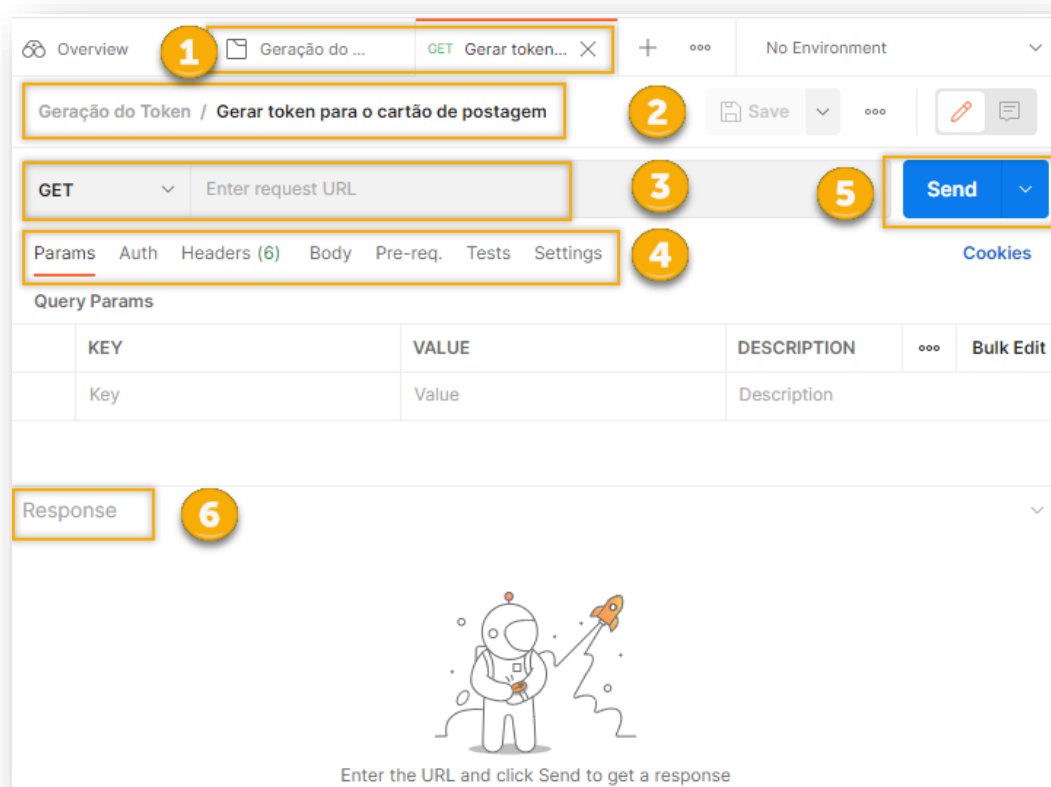
Item 2: Name of the request and which collection it belongs;

Item 3: Verb and the *url base*;

Item 4: Information that can be added by parameters, authorization data, request's head, request's body.

Item 5: Send button, if you press the down arrow, you will be able to send and to download the result;

Item 6: Response.



E. Making the first request through Postman

It is necessary to use the information shown in the [Correios API](#), in order to fill in the request's fields:

Recapitulating, in the previous steps we saw:

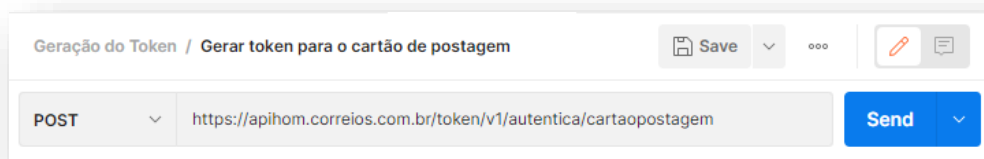
- Access to the components: Login and password to access components;
- Request address ;
- Verb used in the request;
- Request body.

For this first example, the following image shows the (1) main address that will be used, (2) the verb and (3) request address.

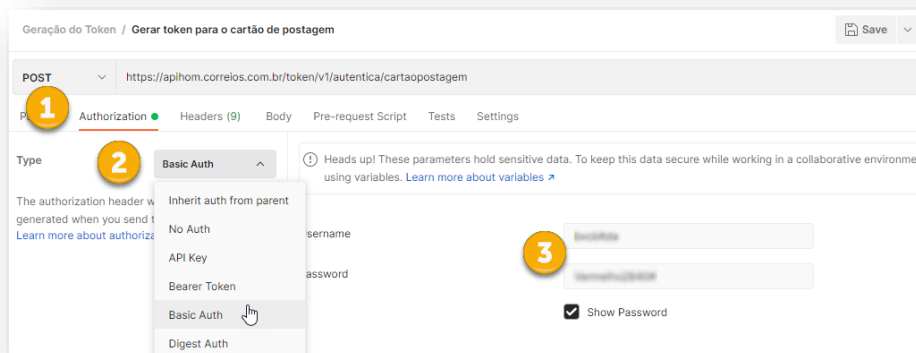


In this way, the address that will be used in the request is the junction of the main address (1) and the request address (3). Example: <https://apihom.correios.com.br/token/v1/autentica/cartaopostagem>.

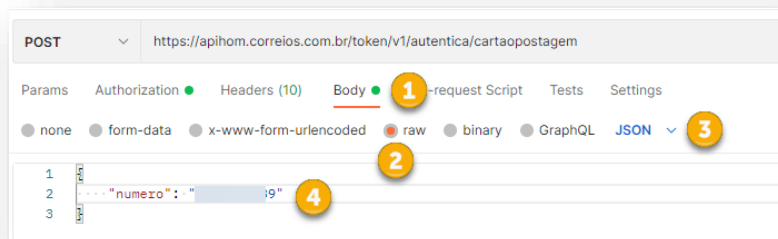
At the request address and at the verb will be like this:



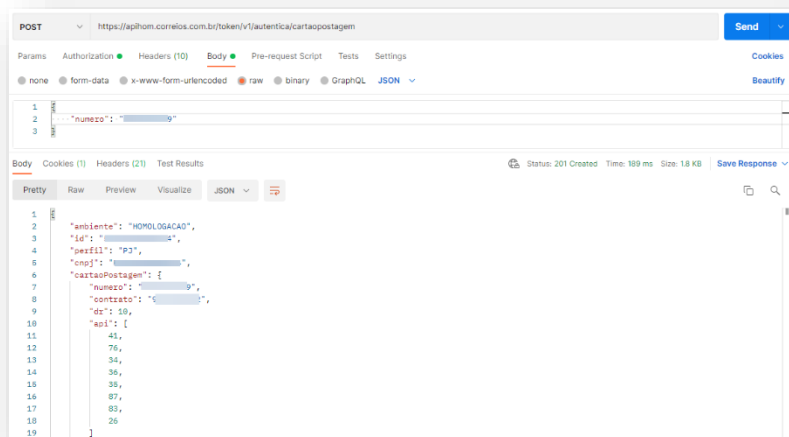
At login and password, click on the tab (1) *Authorization*, select the Type (2) “*Basic Auth*” and (3) fill in the data of login and password:



The next step consists to send the postage card’s data, click on (1) *Body*, (2) *raw*, (3) *JSON* and (4) paste the JSON from Correios Web Services homologation site.



Click on “Send” and see the return:

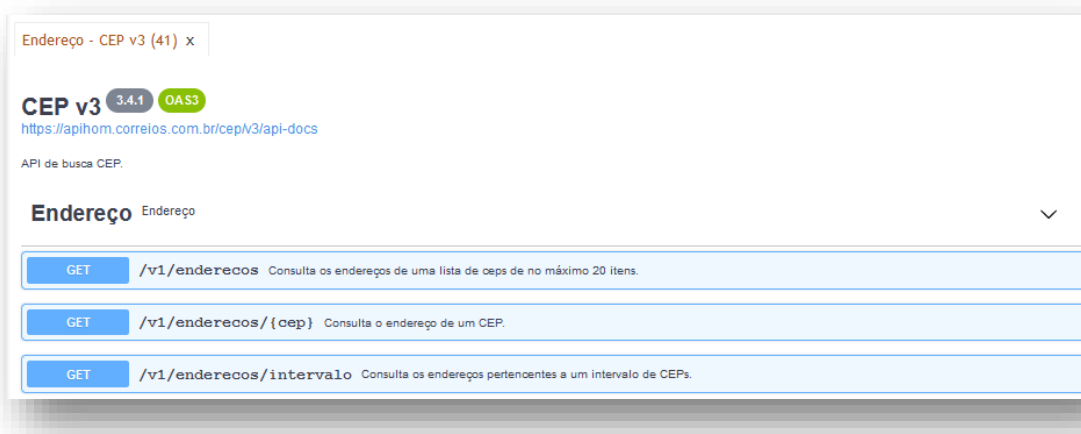


Based on this step-by-step, you will be able to replicate to any API.

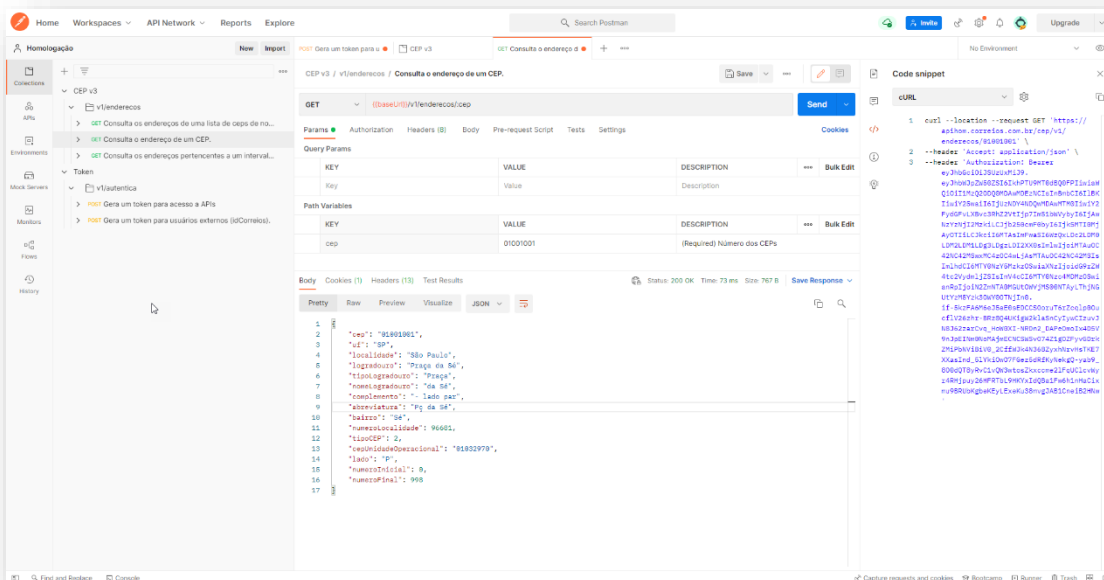
9. API CEP (API Zip Code)

The CEP (Zip Code) is a numerical set consisting of eight digits, which guides and accelerates the forwarding, processing and distribution of correspondence objects, through their assignment to locations, public places, Post Office units, services, public agencies, companies and buildings. (To learn more: [All about CEP](https://apihom.correios.com.br/cep/v3/api-docs))

The tests can be ran through the Correios API page, as follows:



Or through Postman:



In this example, the method that requires the value to query the address of a zip code

will be used.

The base url is: 'https://apihom.correios.com.br/cep/ ' and, according to the documentation, that will be a GET : “/v1/endereços/{cep}”.

Note: By default the zip code is formed by 8 numeric digits, but for some addresses we use the 0 (zero) on the left, example: 01001001.

To the request:

```
curl --location --  
request GET 'https://apihom.correios.com.br/cep/v1/enderecos/01001001' \  
--header 'Accept: application/json' \  
--  
header 'Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJhbWJpZnZW50ZSI6IkhPTU9MT0dBQ  
0FPiIiwiaWQiOiIlMzQ2ODQ0MDAwMDEZNCIsInBmbCI6IlBKIiwiy25waiI6IjUzNDY4NDQwMDAwMTM0I  
iwiY2FydGFvLXBvc3RhZ2VtIjp7Im51bWVwybyI6IjAwnNzYzNjI2MzkiLCJjb250cmF0byI6Ijk5MTIOM  
jAYOTiIiLCJkciI6MTAsImFwaSI6WzQxLDc2LDM0LDM2LDM1LDg3LDgzLDI2XX0sImlwIjoiaMTAuOC42N  
C42MSwxMC4zOC4wLjAsMTAuOC42NC42MSIsImldhdCI6MTY0NzY5MzkzOSwiaXNzIjoiaWoiImTAuOC42N  
mljZSIsImV4cCI6MTY0Nzc4MDMzMzOswianRpIjoiaWoiImNTAyLThjNGUtYm5yZk3O  
WY0OTNjIn0...
```

The response will be:

```
{
  "cep": "01001001",
  "uf": "SP",
  "localidade": "São Paulo",
  "logradouro": "Praça da Sé",
  "tipoLogradouro": "Praça",
  "nomeLogradouro": "da Sé",
  "complemento": "- lado par",
  "abreviatura": "Pç da Sé",
  "bairro": "Sé",
  "numeroLocalidade": 96681,
  "tipoCEP": 2,
  "cepUnidadeOperacional": "01032970",
  "lado": "P",
  "numeroInicial": 0,
  "numeroFinal": 998
}
```

In case of exception, there is the return http 400 from server such as the following examples:

In case of non-existing CEP (zip code):

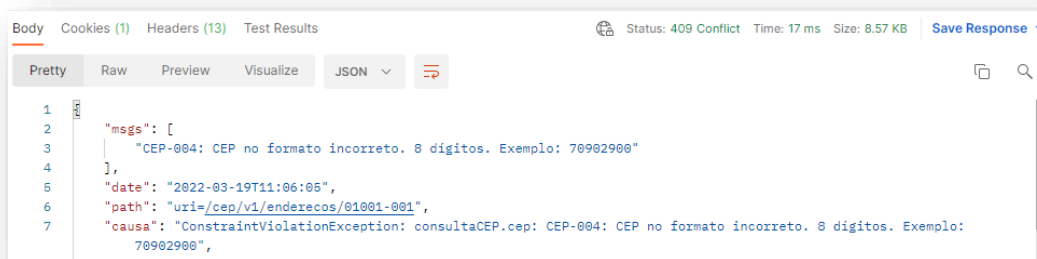


```

1  {
2    "cep": "01001011",
3    "txMsg": "CEP-003: 0 CEP 01001011 não foi encontrado"
4  }
  
```

In case of non-standard typing:

In this example, the CEP: 01001-001 was used, with the hyphen. However, the return informs the number of characters and the example format.

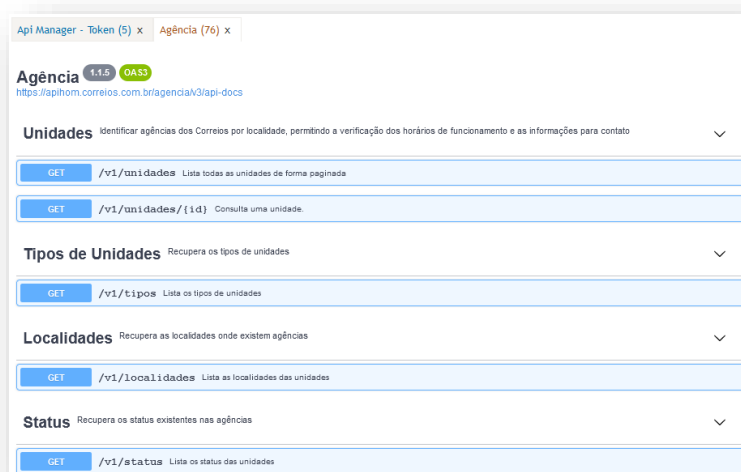


```

1  {
2    "msgs": [
3      "CEP-004: CEP no formato incorreto. 8 dígitos. Exemplo: 70902900"
4    ],
5    "date": "2022-03-19T11:06:05",
6    "path": "/api/v1/enderecos/01001-001",
7    "causa": "ConstraintViolationException: consultaCEP.cep: CEP-004: CEP no formato incorreto. 8 dígitos. Exemplo: 70902900",
8    "errorMessage": "Invalid CEP format. CEP-004: CEP no formato incorreto. 8 dígitos. Exemplo: 70902900"
9  }
  
```

10. API Agência (Agency/Post Office API)

The API Search Agency has the functionality of showing the post office units per location. There are search mechanisms by post-office units or or by location.



To use the units query, it is necessary to retrieve the unit types and status data.

For the Agency API the base url: "https://apihom.correios.com.br/agencia", and for the example a GET will be used: "/v1/unidades" with parameters.

Request example:

```
curl --location --request GET 'https://apihom.correios.com.br/agencia/v1/unidades?status=2&status=5&codigoTipoUnidade=09&codigoTipoUnidade=12&codigoTipoUnidade=43&codigoTipoUnidade=46&codigoTipoUnidade=24&codigoTipoUnidade=25&codigoTipoUnidade=27&codigoTipoUnidade=16&codigoTipoUnidade=18&codigoTipoUnidade=20&codigoTipoUnidade=21&codigoTipoUnidade=24&codigoTipoUnidade=25&municipio=Sao Paulo&bairro=CENTRO&uf=SP&page=0&size=50' \--header 'Accept: application/json' \--header 'Authorization: Bearer eyJhbGciOiJSUzUxMiJ9.eyJhbWVwZW50ZSI6IkhPTU9MT0dBQ0FPIiwiaWQiOiIlMzQ2ODQ0MDAwMDEzNCIsInBmbCI6IlBKlwiY25waiI6IjUzNDY4NDQwMDAwMTM0IiwiaWF0IjE2Y2YyY2YybyI6IjAwNzYzNjIzMzkiLCJjb250cmF0byI6Ijk5MTIOMjAyOTIiLCJkaXI6ImFwaSI6IjZmZDc2LDM0LDM2LDM1LDg3LDgzLDI2XX0sImVwIjoimTAuOC42NC42MSwxC42OC4wLjAsMTAuOC42NC42MSIsImVhdCI6ImTY0NzY5MzkzOSwiaXNjaWoidG9rZW4tc2VydmVjZSIsImV4cCI6ImTY0Nzc4MDMzMzOsSwianRpiIjoimNTA0MGUtOWVjMS00NTAyLTJhNGUyYzMyYzkyOjY0OTNjIn0...'`
```

Response:

```
{
  "itens": [
    {
      "id": "00024419",
      "codigoAntigo": "72300019",
      "codigoCadastroGeral": 16439,
      "nome": "AC CENTRAL DE SAO PAULO",
      "ativa": true,
      "status": "2",
      "descStatus": "INSTALADO",
      "tipo": "A",
      "tipoUnidade": {
        "codigo": "09",
        "descricao": "AGENCIA CORREIO",
        "sigla": "AC -TCO"
      },
      "emails": [
        "spmacacp@correios.com.br"
      ],
      "endereco": {
        "cep": "01031970",
        "uf": "SP",
        "localidade": "SAO PAULO",
        "municipio": "SAO PAULO",
        "logradouro": "PRACA DO CORREIO"
```



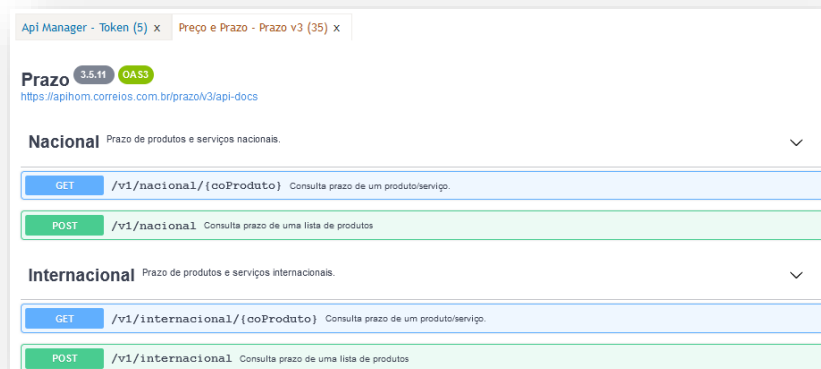
```
        "bairro": "CENTRO",
        "numero": "SN",
        "codigoIbge": "3550308",
        "regiao": "SDE",
        "longitude": "-46.636301",
        "latitude": "-23.544201",
        "fuso": "UTC-03:00",
        "fusoVerao": "UTC-03:00"
    },
    "horarios": {
        "funcionamento": "SEGUNDA À SEXTA",
        "iniExpediente": "09:00",
        "fimExpediente": "18:00",
        "limitePostagemSemana": "17:35"
    },
    "codigoSro": [
        "01009972"
    ],
    "dhAlteracao": "2021-03-16T01:01:22.27"
    }, ...],
    "page": {
        "size": 50,
        "totalElements": 7209,
        "totalPages": 145,
        "number": 0
    }
}
```

In this example search, the size attribute informs how many elements per page, a size=50 was placed in the request, that is, 50 elements per page, totaling 145 pages. If the same query is performed to inform the content of the other pages, just inform the value of 0..144 in the page attribute, if you keep the size=50.

11. API Preço (Price API) and API Prazo (Delivery Time API)

The search for price and delivery time is separated from each other, to meet the concept of micro-services. Thus, it makes two pages in the Correios API catalog.

API Delivery Time (Prazo) (35):



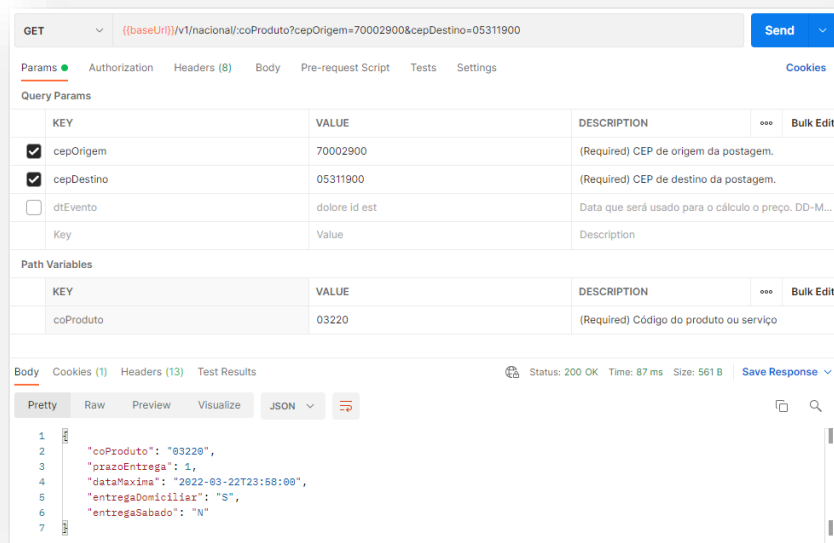
For the url base: <https://apihom.correios.com.br/prazo/> using GET:
/v1/nacional/{coproduto}

[illegible]

We have the following response:

```
{
  "coProduto": "03220",
  "prazoEntrega": 1,
  "dataMaxima": "2022-03-22T23:58:00",
  "entregaDomiciliar": "S",
  "entregaSabado": "N"
}
```

Using Postman:



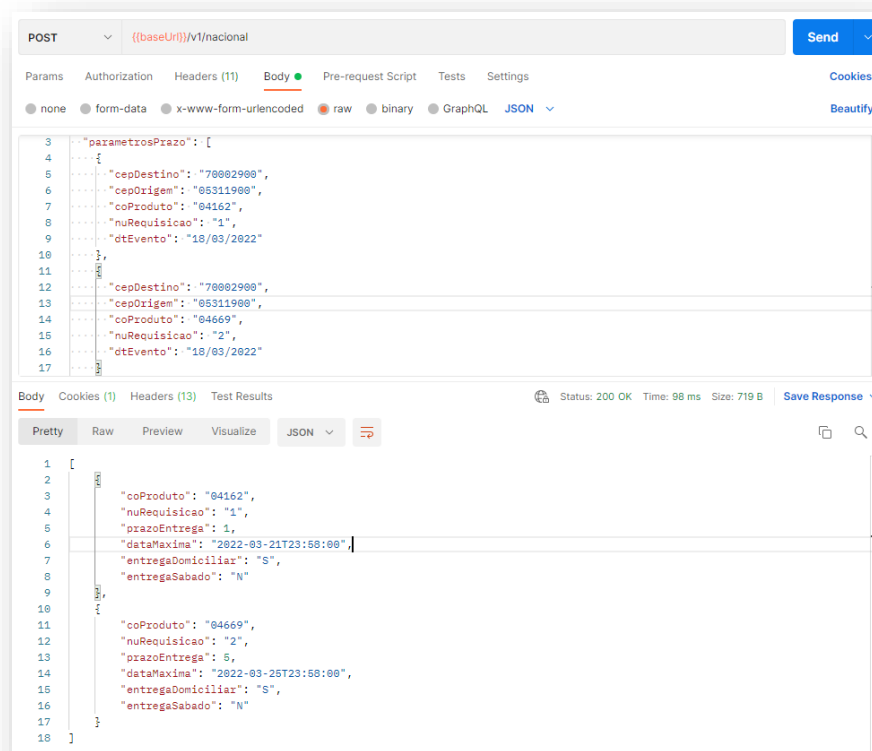
For the url base: <https://apihom.correios.com.br/prazo/> using POST: /v1/nacional

```
curl --location --
request POST 'https://apihom.correios.com.br/prazo/v1/nacional' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
header 'Authorization: Bearer eyJhbGciOiJSUzUxMiJ9.eyJhbWJpZW50ZSI6IkhPTU9MT0dB
Q0FPIiwiBvc3RhZ2VtIjpw7Im51bWVyb3R5I6IjAwNzYzNjI2MzkiLCJjb250cmF0byI6Ij... \
--data-raw '{
  "idLote": "1",
  "parametrosPrazo": [
    {
      "cepDestino": "70002900",
      "cepOrigem": "05311900",
      "coProduto": "04162",
      "nuRequisicao": "1",
      "dtEvento": "18/03/2022"
    },
    {
      "cepDestino": "70002900",
      "cepOrigem": "05311900",
      "coProduto": "04669",
      "nuRequisicao": "2",
      "dtEvento": "18/03/2022"
    }
  ]
}'
```

To obtain the following response:

```
[
  {
    "coProduto": "04162",
    "nuRequisicao": "1",
    "prazoEntrega": 1,
    "dataMaxima": "2022-03-21T23:58:00",
    "entregaDomiciliar": "S",
    "entregaSabado": "N"
  },
  {
    "coProduto": "04669",
    "nuRequisicao": "2",
    "prazoEntrega": 5,
    "dataMaxima": "2022-03-25T23:58:00",
    "entregaDomiciliar": "S",
    "entregaSabado": "N"
  }
]
```

Using Postman



API Price

For the url base: `https://apihom.correios.com.br/preco` we use GET:
`/v1/nacional/{codigoServico}`

For the request:

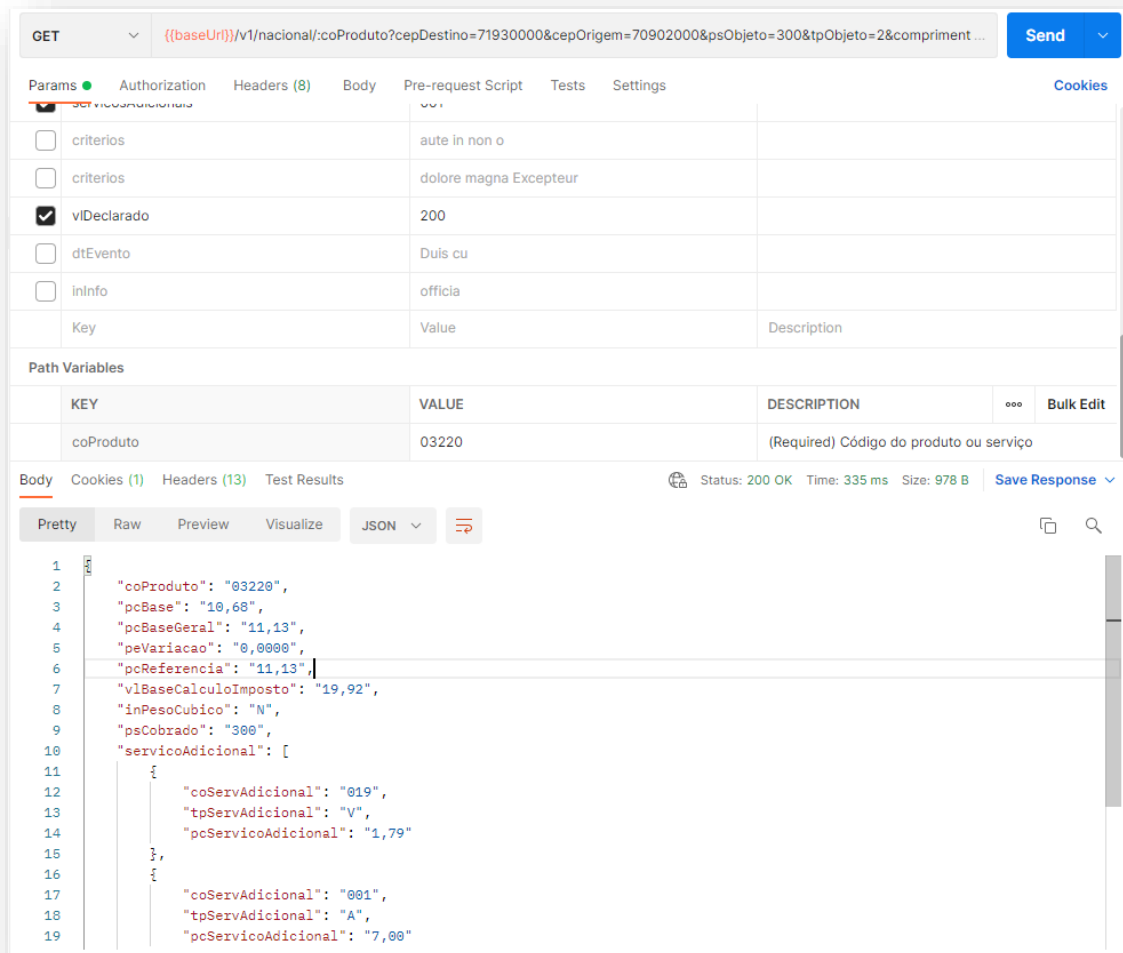
```
curl --location --request GET 'https://apihom.correios.com.br/preco/v1/nacional/03220?cepDestino=71930000&cepOrigem=70902000&psObjeto=300&tpObjeto=2&comprimento=20&largura=20&altura=20&servicosAdicionais=019&servicosAdicionais=001&v1Declarado=200' \--header 'Accept: application/json' \--header 'Authorization: Bearer eyJhbGciOiJSUzUxMiJ9.eyJhbWJpZW50ZSI6IkhPTU9MT0dBQ0FPIiwiaWQiOiIlMzQ2ODQ0MDAwMDEzNCIsInBmbCI6IlBKIIiwY25waiI6IjUzNDY4NDQwMDAwMTM0IiwiyY2FydGFvLXBvc3RhZ2VtIjp7Im51bWVwybyI6IjAwNzYzNjIzMzkiLCJjb250cmF0byI6Ijk5MTIOMjAyOTIiLCJkciI6MTAsImFwaSI6WzQxLDc2LDM0LDM2LDM1LDg3LDgzLDI2XX0sImlwIjoiaMTAuOC42NC42MSwxMC4zOC4wLjAsMTAuOC42NC42MSIsImlhdiCI6MTY0NzY5MzkzOSwiaXNzIjoiaWoidG9rZW4tc2VydmVjZSIsImV4cCI6MTY0Nzc4MDMzMzOswianRpIjoiaWoiNTA0MGUtOWVjMS00NTAyLTJhNGUtYm5yZk30WY0OTNjIn0...'>
```

We have the following response:

```
{
  "coProduto": "03220",
  "pcBase": "10,68",
  "pcBaseGeral": "11,13",
  "peVariacao": "0,0000",
  "pcReferencia": "11,13",
  "vlBaseCalculoImposto": "19,92",
  "inPesoCubico": "N",
  "psCobrado": "300",
  "servicoAdicional": [
    {
      "coServAdicional": "019",
      "tpServAdicional": "V",
      "pcServicoAdicional": "1,79"
    },
    {
      "coServAdicional": "001",
      "tpServAdicional": "A",
      "pcServicoAdicional": "7,00"
    }
  ],
  "peAdValorem": "0,0100",
  "vlSeguroAutomatico": "21,00",
  "qtAdicional": "0",
  "pcFaixa": "11,13",
}
```

```
"pcFaixaVariacao": "11,13",
"pcProduto": "11,13",
"pcTotalServicosAdicionais": "8,79",
"pcFinal": "19,92"
}
```

On Postman:



For pricing up to 5 simulations, we use POST. For the url base: <https://apihom.correios.com.br/preco> and with POST: `/v1/nacional`

```
curl --location --
request POST 'https://apihom.correios.com.br/preco/v1/nacional' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJSUzUxMiJ9.... \
--data-raw '{
  "idLote": "1",
  "parametrosProduto": [
```

```
{
  "coProduto": "04162",
  "nuRequisicao": "1",
  "cepOrigem": "70902000",
  "psObjeto": "300",
  "tpObjeto": "2",
  "comprimento": "20",
  "largura": "20",
  "altura": "20",
  "servicosAdicionais": [{ "coServAdicional": "019"}, {"coServAdicional": "00
1"} ],
  "vlDeclarado": "100",
  "dtEvento": "18/03/2022",
  "cepDestino": "71930000"
}, {
  "coProduto": "04162",
  "nuRequisicao": "1",
  "cepOrigem": "70902000",
  "psObjeto": "300",
  "tpObjeto": "2",
  "comprimento": "20",
  "largura": "20",
  "altura": "20",
  "servicosAdicionais": [{ "coServAdicional": "019"}, {"coServAdicional": "00
1"} ],
  "vlDeclarado": "100",
  "dtEvento": "18/03/2022",
  "cepDestino": "71930000"
}
]
```

We have the following response:

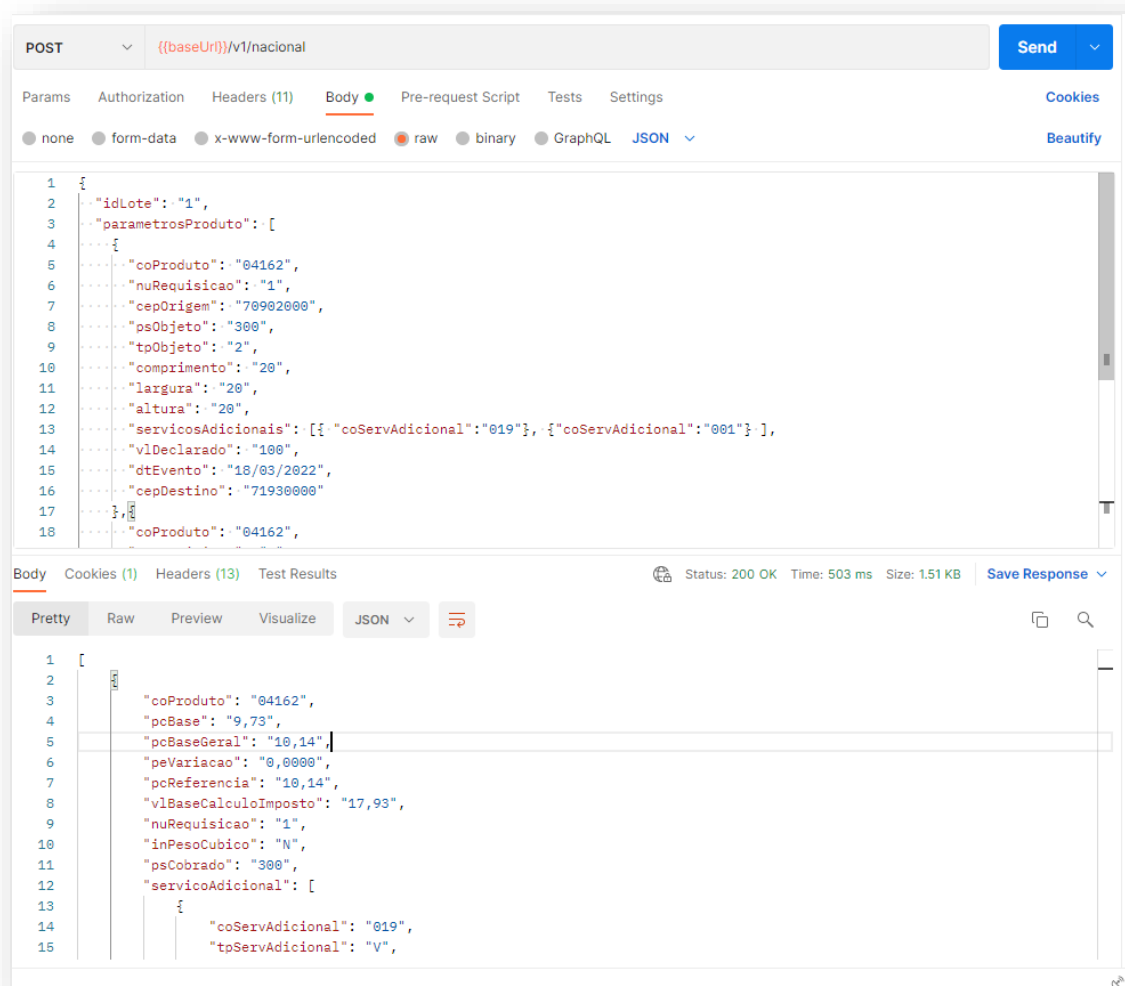
```
[ {
  "coProduto": "04162",
  "pcBase": "9,73",
  "pcBaseGeral": "10,14",
  "peVariacao": "0,0000",
  "pcReferencia": "10,14",
  "vlBaseCalculoImposto": "17,93",
  "nuRequisicao": "1",
  "inPesoCubico": "N",
  "psCobrado": "300",
  "servicoAdicional": [
    {
      "coServAdicional": "019",
```

```

        "tpServAdicional": "V",
        "pcServicoAdicional": "0,79"
    },
    {
        "coServAdicional": "001",
        "tpServAdicional": "A",
        "pcServicoAdicional": "7,00"
    }
],
"peAdValorem": "0,0100",
"vlSeguroAutomatico": "21,00",
"qtAdicional": "0",
"pcFaixa": "10,14",
"pcFaixaVariacao": "10,14",
"pcProduto": "10,14",
"pcTotalServicosAdicionais": "7,79",
"pcFinal": "17,93"
},
{
    "coProduto": "04162",
    "pcBase": "9,73",
    "pcBaseGeral": "10,14",
    "peVariacao": "0,0000",
    "pcReferencia": "10,14",
    "vlBaseCalculoImposto": "17,93",
    "nuRequisicao": "1",
    "inPesoCubico": "N",
    "psCobrado": "300",
    "servicoAdicional": [
        {
            "coServAdicional": "019",
            "tpServAdicional": "V",
            "pcServicoAdicional": "0,79"
        },
        {
            "coServAdicional": "001",
            "tpServAdicional": "A",
            "pcServicoAdicional": "7,00"
        }
    ]
},
"peAdValorem": "0,0100",
"vlSeguroAutomatico": "21,00",
"qtAdicional": "0",
"pcFaixa": "10,14",
"pcFaixaVariacao": "10,14",
"pcProduto": "10,14",
"pcTotalServicosAdicionais": "7,79",
"pcFinal": "17,93"
}]

```


On Postman



12. API Rastro (Tracking API)

This API allows querying the status of the object within the postal flow. The tab of this API will be:



As with the other APIs, a base url is used: `https://apphom.correios.com.br/rastro` with the GET: `/v1/objetos/{codigoObjeto}`

Examples of objects: DG049186230BR, DG049186226BR e DG049186212BR

Making the request:

```
curl --location --
request GET 'https://apihom.correios.com.br/srorastro/v1/objetos/DG049186226BR?r
esultado=T' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer eyJhbGciOiJSUzUxMiJ9...
```

Receiving the response:

```
{
  "versao": "1.8.7",
  "quantidade": 1,
```

```
"resultado": "Todos os Eventos",
"objetos": [
  {
    "codObjeto": "DG049186226BR",
    "tipoPostal": {
      "sigla": "DG",
      "descricao": "ENCOMENDA SEDEX (ETIQ LOGICA)",
      "categoria": "SEDEX"
    },
    "dtPrevista": "2022-03-11T20:58:00",
    "modalidade": "F",
    "peso": 0.5,
    "formato": "Pacote",
    "eventos": [
      {
        "codigo": "BDE",
        "tipo": "01",
        "dtHrCriado": "2022-03-11T17:03:00",
        "descricao": "Objeto entregue ao destinatário",
        "unidade": {
          "tipo": "Unidade de Atendimento",
          "endereco": {
            "uf": "AC",
            "cidade": "XAPURI"
          }
        }
      },
      {
        "codigo": "OEC",
        "tipo": "03",
        "dtHrCriado": "2022-03-11T17:02:45",
        "descricao": "Objeto em rota de entrega",
        "detalhe": "Aguarde envio de link para acompanhar a entrega em tempo real\r\n",
        "unidade": {
          "tipo": "Unidade de Tratamento",
          "endereco": {
            "uf": "SP",
            "cidade": "SAO PAULO"
          }
        }
      },
      {
        "codigo": "RO",
        "tipo": "01",
        "dtHrCriado": "2022-03-11T17:02:35",
        "descricao": "Objeto em trânsito",
        "detalhe": "Por favor, aguarde",
        "unidade": {
```

```
        "tipo": "Unidade de Distribuição",
        "endereco": {
            "uf": "DF",
            "cidade": "BRASILIA"
        }
    },
    "unidadeDestino": {
        "tipo": "Unidade de Tratamento",
        "endereco": {
            "uf": "GO",
            "cidade": "APARECIDA DE GOIANIA"
        }
    }
},
{
    "codigo": "PO",
    "tipo": "01",
    "dtHrCriado": "2022-03-10T19:10:36",
    "descricao": "Objeto postado",
    "unidade": {
        "tipo": "Unidade de Atendimento",
        "endereco": {
            "uf": "SP",
            "cidade": "SAO PAULO"
        }
    }
}
]
}
```

13. Final considerations

This material was developed to make it easier for the reader who was used to the Correios Web Services, which had a SOAP XML standard, and may miss the explanatory material that was provided for each system, with the migration to REST. With the Correios API, we centralized the documentation and as the update is dynamic, it is recommended that you return frequently.

For all APIs mentioned in this material, they are the ones that are released for use. Thus, if you have questions regarding the API, the procedure is to contact the Commercial Assistant of your region or Contract.

Service information can be obtained from our website:
<https://www.correios.com.br/falecomoscorreios/central-de-atendimento>